



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/711,733	09/30/2004	Lee George Laborczfalvi	2006579-0143	5732

69665 7590 04/01/2009

CHOATE, HALL & STEWART / CITRIX SYSTEMS, INC.
TWO INTERNATIONAL PLACE
BOSTON, MA 02110

EXAMINER

ABDUL-ALL, OMAR R

ART UNIT

PAPER NUMBER

2178

MAIL DATE

DELIVERY MODE

04/01/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/711,733

Applicant(s)

LABORCZ FALVI ET AL.

Examiner

OMAR ABDUL-ALI

Art Unit

2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 30 September 2004.
2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-29 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☐ Claim(s) _____ is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
10) ☒ The drawing(s) filed on 30 September 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-8508)
Paper No(s)/Mail Date 12/04, 5/06, 4/08
4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
5) ☐ Notice of Inventor's Patent Application
6) ☐ Other: _____

DETAILED ACTION

The following action is in response to the original filing of September 30th, 2004.

Claims 1-29 are pending and have been considered below.

Claim Rejections - 35 USC § 101

1. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

2. Claims 14-19 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 14-19 are drawn to a computer program per se. A computer program is not a series of steps or acts and this is not a process. A computer program is not a physical article or object and as such is not a machine or manufacture. A computer program is not a combination of substances and therefore not a compilation of matter. Thus, a computer program by itself does not fall within any of the four categories of invention. Therefore, Claims 14-19 are not statutory.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 11-13 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to

which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claim 11 recites "determining that the window handle is associated with the requested window further comprises determining that **no** association between the window handle and the requested one of the virtual window name and the virtual window class identifier exists" (emphasis added). There is no disclosure of a determination that no association exists in the specification.

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 1-13 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 1 recites "associating a window handle with the determined virtual window name". However, the only determination step in Claim 1 is the determination of a literal window name. Appropriate correction is required.

Claim Objections

6. Claim 13 is objected to because of the following informalities: Claim 13 contains the phrase "the operating system". There is insufficient antecedent basis for this limitation. Appropriate correction is required.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 10-12 are rejected under 35 U.S.C. 102(b) as being anticipated by Parker et al. (US 5,781,720).

Claim 10: Parker discloses a method for virtualizing access to windows, comprising:

a. receiving a request to identify one of a virtual window name and a virtual window class identifier, the request received from a process executing within the context of a user account and including a window handle (tag). Parker discloses a test script specifies a request against a logically named LSE (window), and a test executive resolves the LSE's logical name contained in the script command into a GUI specific name as a parameter (column 13, lines 1-25).

b. determining that the window handle (tag) is associated with the requested one of the virtual window name and the virtual window class identifier (column 19, lines 40-50).

c. returning to the requesting process the determined window information (column 13, lines 1-25).

Claim 11: Parker discloses a method for virtualizing access to windows as in claim 10 above, further comprising determining that the window handle is associated with the requested window name further comprises determining that no association between the

window handle and the requested one of the virtual window name and the virtual window class identifier exists (column 23, lines 10-22).

Claim 12: Parker discloses a method for virtualizing access to windows as in claim 11 above, further comprising determining the window handle associated with the requested one of the virtual name and the virtual window class identifier from a mapping table (column 23, lines 10-22)

Claim 20: Parker discloses a method for virtualizing access to windows, the method comprising:

- a. intercepting a receipt, from a requester, to paint a title bar for a window, the title bar including the window name, the request including a window handle (column 10, lines 29-42) Parker discloses a test script reads a windows name through a title bar;
- b. determining that the window handle (tag) is associated with the virtual window name (column 20, lines 29-39);
- c. painting the title bar of the window using the virtual window name (column 20, lines 29-39);
- d. indicating to the requestor that the title bar has been painted (column 22, lines 58-68).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 1-9, 13-17, 19, and 21-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Parker et al. (US 5,781,720) in view of Oppermann et al. (US 6,144,377).

Claim 1: Parker discloses a method for virtualizing access to windows, the method comprising:

a. receiving a request related to a window from a process executing within the context of a user account, the request including a virtual window name (logical name) (column 13, 1-25).

b. determining a literal name (GUI specific name) for the window using a scope-specific identifier associated with a particular isolation scope. Parker discloses a test script specifies a request against a logically named LSE (window), and a test executive resolves the LSE's logical name contained in the script command into a GUI specific name as a parameter (column 13, lines 1-25).

Parker does not explicitly disclose issuing to the operating system a request including the determined literal name. However, Oppermann discloses a similar method that further discloses using operating system function calls to retrieve window names (column 21, lines 1-24). It would have been obvious to one having ordinary skill in the art at the time the invention was made to issue to the operating system a request including a determined literal name in Parker. One would have been motivated to issue

the request to the operating system in order to retrieve user interface elements that are managed by the operating system.

Parker modified by Opperman discloses associating a window handle (tag) with the determined virtual window name (column 19, lines 40-50).

Claim 2: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses receiving a request further comprises intercepting a request relating to a window from a process executing in the context of a user account, the request including a virtual window name (column 13, lines 1-25).

Claim 3: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses receiving a request further comprises receiving a request to find a window from a process executing in the context of a user account, the request including a virtual window name (column 13, lines 1-25).

Claim 4: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses receiving a request further comprises receiving a request to create a window from a process executing in the context of a user account, the request including a virtual window name (column 6, lines 39-63).

Claim 5: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses determining a rule associated with the

virtual window name included in the request and determining a literal name for the window responsive to the determined rule (column 13, lines 1-25).

Claim 6: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses determining a literal name further comprises determining a literal window name using a scope-specific identifier associated with an application isolation scope with which the process making the request is associated (column 133, lines 1-25).

Claim 7: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses associating a window handle further comprises storing the virtual window name in a mapping table associated with a window handle (column 23, lines 11-23).

Claim 8: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Oppermann further discloses receiving from the operating system a response to the issued request (column 21, lines 1-24). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to receive a response from the operating system in Parker. One would have been motivated to receive a response from an operating system in order to retrieve user interface elements that are managed by the operating system.

Claim 9: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further disclose replacing the literal window name in the response with a virtual window name (column 26, lines 1-15).

Claim 13: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Oppermann further discloses returning to the requesting process a response received from the operating system (column 21, lines 1-24).

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to receive a response from the operating system in Parker. One would have been motivated to receive a response from an operating system in order to retrieve user interface elements that are managed by the operating system.

Claim 14: Parker discloses a method for virtualizing access to windows, the method comprising:

- a. a hooking mechanism receiving a request related to a window from a process executing within the context of a user account, the request including one of a virtual window name (logical name) and a virtual window class identifier (column 13, 1-25).

- b. a window name virtualization engine forming one of a literal name for the window and a literal class identifier using one of the virtual window name and the virtual window class identifier received in the request and a scope specific identifier associated with a particular isolation scope. Parker discloses a test script specifies a request against a logically named LSE (window), and a test executive resolves the LSE's logical

name contained in the script command into a GUI specific name as a parameter (column 13, lines 1-25).

Parker does not explicitly disclose an operating system interface issuing a request relating to a window, the request including the one of the formed literal name and the formed literal window class identifier for the window. However, Oppermann discloses a similar method that further discloses using operating system function calls to retrieve window names (column 21, lines 1-24). It would have been obvious to one having ordinary skill in the art at the time the invention was made to issue to the operating system a request including a determined literal name in Parker. One would have been motivated to issue the request to the operating system in order to retrieve user interface elements that are managed by the operating system.

Claim 15: Parker and Olbermann disclose a method for virtualizing access to windows, and Parker further discloses the hooking mechanism intercepts a request selected from a group consisting of finding a window, creating a window, enumerating a window, destroying a window, setting a window name, retrieving a window name, retrieving a window class identifier associated with the window, registering a window class, retrieving information about a window class and unregistering a window class (column 13, lines 1-25).

Claim 16: Parker and Olbermann disclose a method for virtualizing access to windows, and Parker further discloses a mapping table storing an association between a window

handle and one of the virtual window name and the virtual window class identifier (column 23, lines 11-21).

Claim 17: Parker and Olbermann disclose a method for virtualizing access to windows, and Parker further discloses the mapping table is associated with the process (column 23, lines 11-21).

Claim 19: Parker and Olbermann disclose a method for virtualizing access to windows, and Parker further discloses a rules engine comprising a rule determining how the window virtualization engine forms the one of the literal name for the window and the literal class identifier for the window (column 13, lines 1-25).

Claim 21: Parker discloses a method for virtualizing access to windows, the method comprising:

- a. receiving a request, relating to a window class (superclass), from a process executing within the context of a user account, the request including a virtual window class identifier (column 13, lines 1-25).

- b. determining a literal window class identifier using a scope specific identifier associated with a particular isolation scope Parker discloses a test script specifies a request against a logically named LSE (window), and a test executive resolves the LSE's logical name contained in the script command into a GUI specific name as a parameter (column 13, lines 1-25).

Parker does not explicitly disclose issuing to the operating system a request including the determined literal window class identifier. However, Oppermann discloses a similar method that further discloses using operating system function calls to retrieve window names (column 21, lines 1-24). It would have been obvious to one having ordinary skill in the art at the time the invention was made to issue to the operating system a request including a determined literal name in Parker. One would have been motivated to issue the request to the operating system in order to retrieve user interface elements that are managed by the operating system.

Parker modified by Oppermann discloses associating a window handle (tag) with the determined literal window class identifier (column 19, lines 40-50).

Claim 22: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses receiving a request further comprises intercepting a request relating to a window class from a process executing in the context of a user account, the request including a virtual window class identifier (column 13, lines 1-25).

Claim 23: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses receiving a request further comprises receiving a request to find a window from a process executing in the context of a user account, the request including a virtual window class identifier (column 13, lines 1-25).

Claim 24: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses receiving a request further comprises receiving a request to create a window from a process executing in the context of a user account, the request including a virtual window class identifier (column 6, lines 39-63).

Claim 25: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses determining a rule associated with the virtual window class identifier included in the request and determining a literal name for the window responsive to the determined rule (column 13, lines 1-25).

Claim 26: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses determining a literal name further comprises determining a literal window class name using a scope-specific identifier associated with an application isolation scope with which the process making the request is associated (column 133, lines 1-25).

Claim 27: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses associating a window handle further comprises storing the virtual window class identifier in a mapping table associated with a window handle (column 23, lines 11-23).

Claim 28: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Oppermann further discloses receiving from the operating system a response to the issued request (column 21, lines 1-24). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to receive a response from the operating system in Parker. One would have been motivated to receive a response from an operating system in order to retrieve user interface elements that are managed by the operating system.

Claim 29: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, and Parker further discloses replacing the literal window name in the response with a virtual window name (column 26, lines 1-15)).

11. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Parker et al. (US 5,781,720) in view of Oppermann et al. (US 6,144,377) and further in view of Craycroft (US 5,856,826).

Claim 18: Parker and Oppermann disclose a method of virtualizing access to windows as in claim 1 above, but neither reference explicitly discloses a second mapping table associated with a second process. Craycroft discloses a similar system that further discloses maintaining window data in multiple mapping tables (column 7, lines 5-15). It would have been obvious to one having ordinary skill in the art at the time the invention was made to include a second mapping table associated with a second processing

because the use of multiple mapping tables is a known technique in the computer arts. One would have been motivated to include a second mapping table in order to increase efficiency.

Conclusion

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

- a. Alimpich (5,784,057);
- b. Ikemoto (5,515,495);
- c. Hashimoto (5,632,002).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to OMAR ABDUL-ALI whose telephone number is (571)270-1694. The examiner can normally be reached on Mon-Fri(Alternate Fridays Off) 8:30 - 6:00 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong can be reached on 571-272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

OAA
March 23, 2009

/Stephen S. Hong/
Supervisory Patent Examiner, Art
Unit 2178